PC Support Advisor

The Essential Resource for PC Support Professionals

Update 72

Understanding Bus Architectures

PCI, EISA, ISA, VL. What next? 3

Julian Moss starts a two-part article on the ins and outs of PC bus architectures. The concluding half of this article will appear in a future edition of PCSA.

File: H0820.1

Understanding Windows Security

It's not Fort Knox, but it will keep out the curious 11

Mike Lewis offers some tips on how to enhance the security of a Windows-based PC.

File: O0621.1

Supporting OS/2

More on how to keep OS/2 users happy 15

Tim Sipples presents a selection of the most frequently asked questions about IBM OS/2.

File: O0720.7

How To Troubleshoot Stacker 4.0

Yes, it doubles your users' disk space, but take care 21

We answer some of the most common questions that Stac's technical support staff are currently being asked.

File: A0721.1

On This Month's Utility Disk

Assorted utilities for OS/2 2.1

Utilities for improving security of Windows PCs

"Borland woes" - says Robert Schifreen.

In business today, it's essential to have a clear strategy so that you know what you want to do, and how you intend to achieve it. When Borland first arrived on the scene, the company's strategy was to be a big software company that would take on Microsoft and others. Turbo Pascal was, at the time, the best development environment around. Much easier than Microsoft C, which was still using a command-line compiler.

SideKick, too, was a winner, in terms of being an easy-to-use, simple-to-install application that just got on with the job of doing what it did best.

With a few years, Borland was doing extremely well. It had diversified into databases with Paradox, into spreadsheets with Quattro, and even had the nerve to take on Microsoft's domination of the Basic market by launching Turbo Basic.

A combination of bad luck, poor sales, expensive legal battles and a host of other events led to problems. SideKick version 2 was too large and cumbersome for a TSR, but not feature-rich enough for a full PIM. Buying Ashton Tate was expensive and, by the time the revenue from dBASE for Windows started coming in, things were desperate.

Borland needed to re-define its strategy, and it needed some cash in the bank too. So Quattro Pro went to Novell for a cool $145, as did a million Paradox licences.

Borland's new strategy is to concentrate on being the world's leading supplier of database and programming tools. Indeed, it already claims the title. (I'm sure there's another software company that's into databases and programming, though the name escapes me for the minute.)

The key components in Borland's armoury are its C++ compiler, Paradox 5 (just launched), dBASE 5 for DOS, and the newly-launched dBASE 5 for Windows. Er, dBASE for DOS? They spent time and money developing dBASE for DOS? Why bother? And surely selling Paradox and dBASE alongside each other confuses the market.

While Borland's C++ compiler is an excellent product, you don't get rich selling compilers. Even if you manage to get 75% of the world's "top 50" software companies using it, that's only a few hundred sales. Applications is where the money is, not compilers.

But let's excuse this weird behaviour. Let's give Borland the benefit of the doubt. Borland obviously knows what it is doing. Or does it? Why, for example, has Borland launched SideKick for Windows? And why has it set up a Borland Simplifies division, to sell software for homes and kids?

Either the new Borland is serious about being the leading database and programming tools supplier, or it's not. And from what it's doing at the moment, I reckon it's not.

I sincerely hope that I'm proved wrong, and that the corporate customers will be willing to commit their company's data to Borland software. Only time will tell.

PCSA

PC Support *Advisor*

PC Support *Advisor* is a publication of **International Technology Publishing**.

ISSN: 1031-3966

Editor: Robert Schifreen

Editor in Chief: Les Bell

Publisher: Chris Carvan

Email:
pcsa-ed@oakworth.demon.co.uk

International Technology Publishing Inc.
North American Subscriptions
Suite 200
555 De Haro Street
San Fransisco, CA94107
USA

Telephone: (415) 255 1295
Facsimile: (415) 255 8496

International Technology Publishing Inc.
European Subscriptions
13 Callcott Street
London
W8 7SU
United Kingdom

Telephone: (071) 724 9306
Facsimile: (071) 706 3296

International Technology Publishing Pty Ltd.
SE Asian Subscriptions
Level 36 Hong Leong Blding
16 Raffles Quay
Singapore
0104

Telephone: (65) 321 8984
Facsimile: (65) 322 8558

International Technology Publishing Pty Ltd.
Australasian Subscriptions
Suite 511
185 Elizabeth Street
Sydney, NSW, 2000
Australia

Telephone: (02) 261 4683
Facsimile:(02) 261 4741

Understanding Bus Architectures

Julian Moss begins a two-part article on the ins and outs of PC bus architectures.

Not so long ago, a PC was just a PC. You decided on the processor, the clock speed, the amount of RAM needed and the size of the hard disk. Other than those things there wasn't much to choose between them.

Even after the MCA and EISA added an extra element of choice, the type of bus was not a major factor in purchasing decisions unless you were specifying a file server or some other special-purpose system. Today, the ISA bus is as popular as ever. MCA and EISA are still with us, but now there are two additional choices: VL-Bus and PCI.

The graphics and mass-storage performance demanded by modern software means that the type of bus can no longer be ignored. Systems equipped with only an ISA bus aren't adequate for today's applications. So what of MCA and EISA: do they have a future? What are the relative merits of the two local-bus technologies? Which one should your organisation invest in?

In this article, we will look at the currently available PC bus architectures. By the end, you should have a good appreciation of which type of bus is best for your users, and why.

Bus Basics

To most users an expansion bus is a row of slots that you plug cards into. And that's all they need to know. But to appreciate the differences between the types of bus, and their advantages and disadvantages, you need to understand exactly what a bus does. Let's start by describing the features of a computer bus in a generalised way.

A bus carries something from one point to another. A computer bus carries digital information. You can view a computer bus as three separate buses: the address bus, the data bus and the control bus.

The address bus is the set of signal lines which specify the address of a memory or I/O port location which the processor wishes to access. The data bus consists of the signal lines which carry data to or from the CPU. The control bus is made up of the other signals which provide information about the type of bus operation, carry timing signals or requests for an interrupt. Together, they form a channel which connects the processor with its memory and peripherals.

In the first PCs, the bus ran at the same speed as the microprocessor. This is called a synchronous bus. The faster the CPU, the faster the bus and hence the faster the memory and the peripherals it needs. The IBM PC used an Intel 8088 running at 4.77 MHz, so its bus also ran at 4.77 MHz. The first PC ATs ran both the processor and the bus at 6 MHz; later models used 8 MHz. This became a problem as processor speeds increased beyond 8 MHz, because many expansion cards would not work at these higher rates.

Today, the Industry Standard Architecture (ISA) bus, which is what the IBM PC AT bus is now called, is generally run at 8 MHz independent of the speed of the processor. This is termed an asynchronous bus. To distinguish between this and the bus that connects the processor with its memory, and which runs at processor speed, we call the latter the local bus.

The expansion bus is connected to the local bus by a bridge. The bridge takes care of the difference in speed between the processor and the plug-in devices by using latches and buffers to hold information received from or destined for the CPU. When accessing devices over the expansion bus, the processor must sit idle for a number of clock cycles or wait states. This is not good for performance, of course.

The bus itself is controlled by logic circuits which are known generically as the bus controller. The bus controller is one of several functions integrated into the two or three integrated circuit chips present in every PC which are collectively known as the chip set.

"The problem with ISA bus mastering is that there is no arbitration on the bus. The master device has complete control until it decides to relinquish it. It must, therefore, take complete responsibility for the system, including driving the memory refresh signal at the correct interval."

Simplified Signals

Though the design of the ISA bus borrows heavily from the signals used by the Intel 80x86 range of processors, certain features were intended to make system design easier. One example is the provision of multiple interrupt request lines (IRQs) where the processor has only two interrupt inputs.

The bus can reduce the number of signal lines needed by carrying two or more pieces of information over the same lines at different times. This is called multiplexing. An example of this is found in the 8088 processor, in which the lowest eight address lines were used for input or output of data during the second part of a read or write cycle. Another example is the PCI expansion bus connector: data and address information are multiplexed on the same set of contacts, allowing the connector to be made smaller.

When bus signals are only valid for a short period of time, additional signals are used to indicate when they are valid. For example, in the ISA bus the BALE (Buffered Address Latch Enable) signal indicates when the address lines hold a valid address.

On the ISA bus the first 20 address lines are latched. A latch is a circuit which loads and retains a digital value on receipt of an external trigger. In this case latches hold the value of address lines on the bus so that the information is available to expansion cards regardless of the state of the processor's address lines.

Another function performed by the expansion bus is buffering, by which we mean electrical buffering. A digital circuit can drive only a limited number of other circuits. Where the loading of a signal line may exceed the allowable limit - and this is possible in the case of an expansion bus where a variety of different devices could be plugged into slots - buffers must be used.

A buffer is essentially a circuit, the output of which replicates its input, but which can support a larger number of loads. A disadvantage of buffering is that it introduces a propagation delay: the time taken for the input signal to appear at the output.

"The graphics and mass-storage performance demanded by modern software means that the type of bus can no longer be ignored. Systems equipped with only an ISA bus are not adequate for today's applications."

This time is extremely small but it becomes increasingly significant at higher circuit speeds. This is why buffered VL-Bus implementations, though able to support a greater number of expansion cards than unbuffered designs, are limited to a speed of 33 MHz.

Data Transfer

The main function of an expansion bus is the transmission of data between input/output devices and the main memory of the machine. In fact, IBM called the bus the "I/O Channel".

A computer bus supports two types of addressable space: I/O ports and memory. Data is read from, and written to, locations in this space using a variety of methods: the transfer rate achieved depends on the method used. Every type of expansion bus has a theoretical maximum data transfer rate which is dependent on the speed of the bus and its data bandwidth - 8-, 16-, 32- or 64-bits - but because of system overheads the amount of data moved in practice will always be less than this.

Consider a memory read cycle. The processor places on the address bus the address of the memory location it wants to read. It then sets some control signals to indicate that the address is of a memory location and that the access is a read. The memory controller responds to this by placing the requested data on the data bus, from where it can be read into one of the CPU registers.

In this example, the transfer is controlled by the CPU. This is bad from a performance point of view because it ties up the processor, preventing it from carrying out other tasks. If data is being transferred between an I/O device and memory, there is no need for the CPU to do all the work. Therefore most bus designs implement features such as direct memory access (DMA) and bus mastering which allow block transfers to take place autonomously.

An ordinary bus access involves two bus cycles: one for the address, another for the memory read or write. Because many bus transactions involve transferring blocks of data, time can be saved by eliminating the need for a memory address to be given for each access. The more advanced PC buses implement this type of data transfer. Essentially, one address is given, and then a burst of data is read or written, beginning at the given address.

Autonomous Processes

All PC expansion buses provide ways to transfer data between devices and main memory without the processor being involved. Direct Memory Access is the method provided in the earliest IBM PCs. Instead of the processor controlling the transfer, it is performed under the control of a chip called the DMA Controller.

Using DMA, a program requests and is granted a DMA channel, and then sets up a start address and a count of the number of bytes or words to be transferred. The DMA controller copies the data into or out of memory, incrementing the address and decre-

Bus Architectures

"To most users, an expansion bus is a row of slots that you plug cards into, but to appreciate the differences between the types of bus, and the advantages and disadvantages, you need to understand exactly what a bus does."

menting the count, until the count is exhausted. During a DMA memory access the DMA controller, not the CPU, has control of the bus.

DMA is an autonomous process intended to relieve the processor of the often time-consuming task of transferring data from one place to another. A second form of autonomous process implemented in the more advanced types of bus, and exploited by the more high-performance peripherals, is bus mastering.

While DMA performs a specific task - transferring data - and is a function of logic associated with the bus, with bus mastering the active device is an expansion card: the master. A bus master is given complete control of the bus. It can select another device - a slave - and interact with it in exactly the same way as the processor can.

Bus mastering devices are used to achieve rapid data transfer without loading the main processor: bus masters are intelligent devices in their own right. Typical uses include network cards and SCSI host adapters. But, at least in theory, bus mastering could be used to implement a multi-processor system, by plugging multiple processor cards into the expansion bus.

The main memory of a PC uses dynamic RAM (DRAM). DRAM is volatile even when power is applied, and has to be "refreshed" by being read at intervals of 10 - 15 microseconds. This refresh function is carried out by the bus.

In an IBM-compatible PC, the memory refresh cycle is driven by output 1 of the PC's programmable timer chip. While it occurs, other memory accesses cannot be made. So refresh can be regarded, like DMA and bus master memory accesses, as another autonomous process that uses the bus.

With several processes potentially able to share the bus, some form of mechanism is needed to ensure that every device gets its turn, and that high priority and vital tasks like the memory refresh aren't locked out. This mechanism is called arbitration. Various arbitration methods are used: the degree of sophistication varies between the different types of bus.

Other Problems

Most high-performance PCs today incorporate a memory cache of high speed static RAM to reduce the effect that slow DRAM has on the performance of the faster processors. In addition, 486-class processors have their own on-chip cache. This causes a problem for bus designers.

Data held in the memory cache should always mirror the contents of locations in main RAM. Where the processor is the only agent accessing the memory, it isn't difficult to keep things in step. But when autonomous processes can change the contents of RAM without the CPU knowing about it, things start to become rather complex.

To overcome this problem the cache controller and the bus controller have to collaborate, in a process called bus snooping. Whenever a bus master or DMA process accesses memory, it must first check the cache using a "snooping" cycle. If the block of memory to be accessed is currently held in the cache and there is a discrepancy between the two - the cache is said to be dirty - the process must pause whilst memory and cache are resynchronised.

Timing is an important consideration on an expansion bus. Many bus signals are transient in nature and can only be read or asserted at particular points in a bus cycle. Small differences in signal timing between what is expected and what actually happens can be the cause of incompatibilities between peripherals and motherboards. And it's harder for designers to eliminate these differences as bus speeds increase.

The relative merits of a type of bus depend on a number of factors. Performance will be affected by the bus speed and the data bandwidth. But it will also depend on the types of data transfer mechanism offered by the bus, and whether they are exploited by expansion boards and the hardware that drives them.

However, performance is not the

"The ISA bus is basically the expansion bus or I/O channel of the IBM AT, and inherits many of the limitations of the AT's 80286 processor. It has only 24 memory address lines, restricting the range of addressable memory to 16 MB."

only criterion of importance. When choosing the most appropriate bus for an application you should also consider the cost, reliability, compatibility, vendor support and ease of configuration. Let's look at specific features of the five PC expansion buses in more detail.

ISA

The Industry Standard Architecture bus is basically the expansion bus or "I/O Channel" of the IBM AT, and inherits many of the limitations of the AT's 80286 processor. It has only 24 memory address lines, restricting the range of addressable memory to 16 MB. And only 8- and 16-bit data accesses are supported.

20 latched address lines, SA0 to SA19, are provided on the main part of the expansion slot, which corresponds to an 8-bit PC/XT slot. The remaining four address lines, LA20 to LA23, which allow access to extended memory, are part of the 16-bit extension slot. Address lines 17 to 19 are duplicated in this part of the expansion slot. These address lines are unlatched, and so present their information a little ahead of SA0 to SA19.

Address lines LA17 to LA23 are used in a crude system to determine whether an expansion board can accept 16-bit data transfers. LA17 to LA23 are decoded by an expansion board when the signal BALE is raised. Their value indicates the region that a memory access is destined for, truncated to a multiple of 128 KB. If a 16-bit board contains ROM or RAM in this region, it signals that it can accept a 16-bit data transfer using the signal -MEM CS16. The bus controller then knows it can transfer a 16-bit value in one bus cycle, instead of splitting it in half and taking two.

"The expansion bus is connected to the local bus by a bridge. The bridge takes care of the difference in speed between the processor and the plug-in devices by using latches and buffers to hold information received from, or destined for, the CPU."

The result is that you cannot mix 8- and 16-bit cards on an ISA bus if they contain ROM or RAM within the same 128 KB region. If an 8-bit network card has a small RAM image starting at D0000h, for example, it falls within the same region as a 16-bit VGA card's ROM at C0000h. When the system reads from or writes to the network card the VGA card will decode LA17 to LA23 and assume the memory access is meant for it. It will assert -MEM CS16 and the data will be transferred as a 16-bit value. The network card cannot handle this and the system will crash.

Some VGA cards avoid the problem. Either they sense the presence of an 8-bit card in the same memory region and switch to 8-bit operation, or they possess a jumper that can be used to force the card into 8-bit mode. Another solution is to install the VGA card in an 8-bit slot: it cannot see the address lines LA17 to LA23 nor -MEM CS16 and so it is treated as an 8-bit card.

The signals -SMEMR, -MEMR, -SMEMW and -MEMW are used to instruct a memory device to write or read data to or from the data bus. Signals -SMEMR and -SMEMW are on the main part of the expansion connector and only active when the memory address is less than 1 MB. Signals -IOR and -IOW perform a similar function for I/O ports.

Although the 80x86 can support up to 64 KB of I/O address space (0 to FFFFh) the ISA bus decodes only addresses 0 to 3FFh, and I/O ports repeat at 1 KB intervals. (A few ISA implementations may decode I/O addresses 0 to 7FFh, but usually the top five address lines are ignored.) The -I/O CS16 signal is used by devices that support 16-bit I/O transfers.

Memory in a 16-bit system is arranged in banks 16-bits wide. Physical memory accesses are always for 16-bit words starting at an even-numbered byte. Where a byte of data is transferred to or from an odd-numbered byte location, the signal SBHE (Bus High Enable) is used to indicate that the data is on the upper eight bits of the data bus.

Word (two byte) data transfers starting at an odd-numbered address take two memory cycles. This is clearly undesirable from the point of view of performance. Fortunately, most software is generated using a compiler which ensures that data items begin at even-numbered locations.

"The DMA Controller is based on an Intel 8237A chip. This runs at half bus clock speed. Each word takes four DMA clock cycles or eight bus cycles to transfer, excluding overhead at the initiation of the transfer."

Bus Architectures

"The bus is controlled by logic circuits which are known generically as the bus controller. The bus controller is one of several functions integrated into the two or three chips present in every PC which are collectively known as the chip set."

Other bus signals of interest include OSC, an oscillator which runs at 14.31818 MHz. Originally, OSC was the master oscillator for the entire PC: divide it by 3 and you get 4.77 MHz, which is the speed the PC ran at. This frequency was originally chosen because it is used to generate the colour burst signal needed by American colour televisions: one add-in for the original PC was a TV display adapter. OSC is a frequency standard which can be used by devices plugged into the bus.

CLK is the bus clock, which operates at a nominal 8 MHz. In most ISA implementations it is derived by division from the processor clock, so in a 25 MHz or 33 MHz system it is actually 8.33 MHz (25 MHz / 3 or 33.3 MHz / 4), on a 40 MHz system it is 8.0 MHz (40 MHz / 5) and so on. If the system board or BIOS allows you to change the divisor then it is possible to run the bus at a higher speed if the expansion cards will accept it.

-I/O CHCK (I/O Channel Check) is used by expansion cards to indicate that a fatal error has been detected. I/O CHRDY (I/O Channel Ready) is for use by slow devices to lengthen memory or I/O cycles by an integral number of clock cycles. The RESET line allows cards to trigger a hard reset of the system.

Interrupt Lines

An 80x86 processor has only two interrupt inputs, one maskable and one non-maskable. However, the PC AT architecture provides 15 separate maskable IRQ lines. This is achieved using two programmable interrupt controller chips (see "Understanding IRQs" in PCSA Update 69, article H0814.1). Having a number of separate IRQ lines simplifies the design of both expansion boards and the software needed to drive them.

The ISA bus provides access to 11 of these interrupt request lines (IRQ3 - IRQ7, IRQ9 - IRQ12, IRQ14 - IRQ15). Since few ISA machines have more than eight expansion slots, this should be more than adequate. In practice this is not always the case because manufacturers of expansion boards may not allow the use of all the available options.

The ISA bus does not allow devices to share interrupts. The reason is that ISA uses edge-triggered interrupts. When a device requires an interrupt, it pulses the IRQ line high. The interrupt controller responds to the leading edge of the IRQ signal. Unlike a level-sensing system, in which the interrupt line is maintained high until the interrupt has been serviced, there is no way for a device that shares an IRQ line to know if the other device's interrupt has been serviced or not, and whether it is safe to interrupt the system.

Edge-triggered interrupts are simpler to implement than level-sensing interrupts, but theoretically less reliable. Because the interrupt signal is transient, it's possible that the interrupt controller could fail to detect it. Conversely, noise induced on to an IRQ line could be mistaken for an interrupt. In practice, though, these faults should not occur often.

DMA

The ISA bus provides eight channels for direct memory access. For compatibility with the 8-bit PC bus, the first four DMA channels are 8-bits wide. Channel 0 is reserved for the memory refresh. A second DMA controller is used to give an additional four 16-bit channels. Channels 5, 6 and 7 are available for use by devices; channel 4 is used to cascade the 8-bit channels through to the CPU.

The DMA controller is based on an Intel 8237A chip. This runs at half bus clock speed. Each word takes four DMA clock cycles or eight bus cycles to transfer, excluding overhead at the initiation of the transfer. At 8 MHz this works out to a transfer rate of only 1 MB/sec for 8-bit transfers and 2 MB/sec for 16-bit transfers.

Besides its low speed, another limitation of ISA DMA is that data transfers must take place entirely within one memory "page". In the context of DMA, a page is a 64 KB region starting at an address that is an exact multiple of 64 KB in the case of 8-bit DMA, and a 128 KB region starting at an exact multiple of 128 KB for 16-bit DMA.

The reason is that the 8237A DMA controller has only 16 address lines. To enable any region within the 16Mb of address space to be accessed, the most significant bits (bits 17 upwards) of the address are set by writing an appropriate value to the DMA channel's page register. These high address bits are fixed for the entire DMA operation, even if the lower order address bits overflow during the course of the transfer.

The starting (base) address within the page is set by writing it to the base address register for the DMA channel to be used. This sets the lower 16-bits of the address. The number of bytes or words to be transferred is written to the word count register. This is also only 16-bits wide. Consequently a single DMA transfer can move a maximum of 64K bytes or words, and only then if the base address starts at a DMA page boundary.

For 16-bit DMA the base address outputs are multiplied by two by shifting them one bit to the left. This means that the starting point of a 16-bit DMA transfer must be an even-numbered address.

"When bus signals are only valid for a short period of time, additional signals are used to indicate when they are valid. For example, in the ISA bus, the Buffered Address Latch Enable signal indicates when the address lines hold a valid address."

To initiate a DMA transfer the DMA controller registers must be programmed as described. The I/O device then requests control of the bus by raising one of the signals DRQ0 - DRQ3 or DRQ5 - DRQ7. These signals are prioritised - DRQ0 having the highest priority - and correspond to the three 8-bit and three 16-bit DMA channels available for use.

When the processor disengages from the bus it asserts AEN (Address Enable), which prevents other devices from responding to bus signals. The -DACK signal is then activated to acknowledge that the DMA controller has control of the bus. The device responds by lowering the DRQ.

The DMA controller now has full control of the address bus, the data bus and the memory and I/O read and write control lines, and performs the data transfer. Completion of the DMA transfer - when the counter has counted down to zero - is signalled by a pulse sent on the T/C (Terminal Count) signal line.

In theory, the ISA bus allows several DMA channels to be used at once. In some bus implementations, however, simultaneous DMA does not work. This may affect the operation of some software, such as backup programs which use one DMA channel to copy data from disk to memory while another simultaneously copies it out to disk or tape.

Bus Mastering

ISA also supports a limited form of bus mastering. This is invoked using a similar mechanism to DMA. A DRQ signal is used to request control of the system, and is acknowledged by a -DACK in the usual way. When the acknowledgement is received, the bus master device pulls the -MASTER signal low. This gives it control over the entire bus.

The problem with ISA bus mastering is that there is no arbitration on the bus. The master device has complete control until it decides to relinquish it. It must, therefore, take complete responsibility for the system, including driving the memory refresh signal at the correct interval to ensure that the contents of dynamic RAM are not lost. Because of this the IBM documentation recommends that a bus master should not retain control for more than 15 microseconds.

Two bus cycles are the minimum needed for a bus read or write, which gives a maximum theoretical data transfer rate, when 16-bit memory accesses are used, of 8 MB/sec. This is not sustainable for large volumes of data, however. In practice, the maximum throughput of the ISA bus is closer to 4 MB/sec, which is slow when compared to present-day mass storage devices and other peripherals.

Plug And Play ISA

Another inadequacy of the ISA bus is that there is no support for automatic configuration, and the resolution of IRQ, DMA, I/O and memory address conflicts. This is one weakness that has been addressed in later bus designs. Help is at hand, though, in the form of the Plug and Play ISA specification coordinated by Intel and Microsoft.

Plug and Play ISA cards will contain a small ROM containing information such as the type, make and model of the card and the configuration options that are supported. There will also be logic enabling the board to be configured under software control. For fully automatic configuration a system with a plug and play enhanced BIOS will be needed.

At start-up, the BIOS code will interrogate the ROMs, allocate resources after resolving any conflicts, and configure the cards. The configuration settings are stored on the hard disk: those devices needed during bootup will power up active using default settings.

It is too early to say whether all future ISA expansion boards will include the additional hardware needed for Plug and Play. However, the existence of the Plug and Play specification suggests that the PC industry expects the ISA bus to continue to play an important role, despite its other failings, for years to come.

Micro Channel Architecture

MCA was the first attempt at replacing the ISA bus, appearing in 1987. Technologically it is far superior. It addresses the main problems of ISA as they were seen at that time, doubling the data bandwidth to allow for 32-bit processors, and offering improved reliability, switchless setup capabilities and lower electromagnetic interference levels.

MCA is the expansion bus used in most IBM PS/2 systems. The large number of customers committed to buying IBM hardware provided a market for the bus. But only a few other PC manufacturers have adopted it. Most did not want to pay the royalties demanded by IBM for use of its patented technology.

Another reason for the lack of enthusiasm for MCA was that the limitations of ISA were not generally regarded as limitations at that time. ISA was still fast enough for the peripherals that were generally used, reliability and interference were not perceived to be a problem by many users, and most had become used to the idea of fiddling with jumpers to

Bus Architectures

"All PC expansion buses provide ways to transfer data between devices and main memory without the processor being involved. Direct Memory Access is the method provided in the earliest IBM PCs. Instead of the processor controlling the transfer, it is performed under the control of a chip called the DMA Controller."

configure boards. On the other hand, the MCA expansion slot was completely incompatible with existing ISA add-ins, forcing users to buy new MCA boards.

The MCA bus connector helps to achieve lower electromagnetic radiation levels. Every fourth signal on the connector is either an electrical ground or an RF-decoupled power rail. This makes FCC certification easier to achieve - an important consideration in the US.

The bus signals are arranged so that 8-16- and 32-bit slots are possible. 8-bit signals occupy the first 46 pairs of contacts, additional signals for the 16-bit bus take the next 12 pairs, and 32-bit signals occupy a further 31 pairs of contacts. 8- and 16-bit MCA systems have 24 address lines giving an address range of 16 MB, identical to that of the IBM AT. An additional 8 address lines are present in the 32-bit part of the connector. The extra address lines can be disabled if required for AT compatibility by asserting the Memory Address 24 signal.

At the time MCA was designed, the speed of the existing ISA bus was not perceived to be its major weakness. MCA runs at 10 MHz, compared to ISA's 8 MHz, and still requires two bus cycles to complete a data transfer. There is therefore only a 25% improvement in performance when using 16-bit programmed I/O. The main increase comes from the fact that 32-bit data transfers are possible: this will result in 2.5 times the throughput of ISA, but only if 32-bit software is used.

MCA has a software configuration feature called Programmable Option Select (POS). The system configuration is stored on a Reference Disk - this is usually a floppy, though later PS/2s keep the information on a specially reserved area of the hard disk - with the most important details held in battery backed CMOS RAM. The Reference Disk software allocates IRQs, I/O and memory addresses automatically, ensuring that there are no conflicts. MCA is almost plug and play.

Each MCA slot has dedicated signal lines that are used during setup. The Card Setup lines are used to allow the configuration data stored on a board to be accessed. The Card Selected Feedback lines enable boards to communicate with the setup software.

There are some differences from ISA in the way memory and I/O operations are signalled on the MCA bus; however, the end result is the same. The default data transfer is 8-bit. If a 16-bit transfer is possible the Card Data Size 16 signal is asserted; if this is acceptable the card responds using Data Size 16 Return. A similar pair of signals is used for 32-bit transfers. This is a much better mechanism than that used by ISA which I described earlier.

Memory is arranged in banks of 16-bit words or, in the case of a 32-bit systems, 32-bit words. When 8- or 16-bit data transfers occur, the four Byte Enable lines (0 - 3) are used to show which bytes of the data bus carry valid data.

On the MCA bus IRQs correspond to those present in an IBM AT. However, the interrupt lines are level-sensitive: the line remains active while the interrupt is being processed. This makes it possible for MCA devices to share interrupt levels (if the software will support it) as well as providing immunity from noise-triggered false interrupts.

If an expansion card detects a fatal error condition, it can signal this to the bus using the Channel Check signal. The Channel Ready line is used by a device when it needs extra bus cycles to complete an operation.

Faster Transfers

Over MCA the fastest data transfers use burst mode. This is broadly similar to ISA's DMA, though with much better performance. Small blocks of data can be transferred independently of the CPU, taking two bus cycles per transfer. This equates to a peak transfer rate of 20 MB/sec.

MCA also has better support for bus mastering. When a bus master wants control of the bus, it asserts the signal Preempt. The bus arbitration controller responds using the signal Arbitrate/Grant, which indicates that devices wishing to take control of the bus can commence the arbitration process.

Each bus mastering device is assigned a priority level when it is set up: the POS software ensures that no two devices are given the same priority. The devices send their priority levels using the four Arbitration signal lines (0 to 3). When a competing device sees a higher priority level being asserted, it backs off: the device with the highest priority "wins" the arbitration. The Burst signal is used by the winning device to retain control of the bus while it performs its data transfers.

Future Developments

In practical terms, MCA is capable of about four times the data transfer rate of ISA, when 32-bit software is used. Its disadvantage is that expens-

ive MCA boards must be used throughout, even for devices where there is no need for this level of performance. And it is still too slow for the demands of modern applications using graphics, video and audio.

The MCA specification allows for future enhancements. A faster method of data transfer called streaming data mode can be used. This needs only a single bus cycle per unit of transfer, and can handle larger blocks of data than burst mode. This would push maximum throughput to 40 MB/sec.

The MCA specification also allows for the bus clock to be increased up to 20 MHz, though whether existing expansion cards would work at this speed is debatable. Provision has also been made for 64-bit operation, by multiplexing an additional 32-bits of data on to the 32 address lines. If all these enhancements were combined, MCA would be capable of up to 160 MB/sec, making it a match for the capabilities of local bus.

These enhancements are not supported by current MCA products. It remains to be seen, in the light of the development of local expansion buses, whether there will be a demand for faster MCA products in the future.

EISA

Extended Industry Standard Architecture was the rest of the PC industry's response to IBM's MCA. Not wishing to be beholden to Big Blue, a group of nine manufacturers headed by Compaq worked together to develop an open standard for a 32-bit expansion bus which would provide similar performance to MCA, with the added benefit that it would be compatible with existing ISA expansion boards.

EISA is two buses in one. The key to its backward compatibility is the use of a specially-designed two-tier connector. The top tier is a standard ISA connector. Beneath it is the EISA connector with 90 extra contacts sandwiched between the old ISA bus signals. Keys in the base of the connector prevent ISA cards from being pushed all the way in, where they would short out the narrower EISA signal contacts. Another compatibility factor is the retention of the ISA bus standard speed of 8.33 MHz.

The EISA connector includes 16 new data lines giving a 32-bit data path, plus an additional eight address lines which increase the memory address range to 4 GB. Memory is addressed as an array of 32-bit words, so the bus includes four Byte Enable signals BE0 - BE3 to indicate which bytes are valid for a data transfer.

EISA cards may support 8-, 16- or 32-bit transfers. When a data access is made, a 16-bit card signals that it can accept a 16-bit transfer using the EX16 signal. A 32-bit card uses EX32. The EISA bus controller can convert data between one width and another to match the capabilities of different devices.

Like MCA, EISA boards are software configurable using the EISA Configuration Utility (ECU). Setup data is stored in the PC's battery-backed CMOS RAM.

The EISA bus has a slot-specific signal AENx, where x is the slot number. This signal is used to enable each slot individually and allow card configuration data to be accessed. The AENx lines make it possible for two or more identical boards to be installed and separately configured.

Each EISA slot has its own unique block of I/O addresses x000h - xFFFh. This makes I/O conflicts impossible. Each card carries configuration data which is accessed through a block of I/O addresses. This information includes a product identification number which identifies the manufacturer and type of board, as well as details of the configuration options supported.

EISA devices can share IRQs. The interrupt request lines are programmable. When used by EISA boards they are level-sensing; when used by an ISA card they are edge-triggered and sharing is not possible.

Standard EISA data transfers require two bus cycles. EISA also allows compressed data transfers, where only 1.5 cycles per unit of data are needed.

The fastest transfer method is burst mode. In burst mode, only one bus cycle is needed for each data transfer, giving a peak transfer rate of 33 MB/sec. However, only the first 10-bits of the memory address can change when burst mode is used, so transfers are restricted to blocks of 1 KB or less.

Bulk Transfers

EISA has the same number of DMA channels as ISA, though all the EISA DMA types support 32-bit as well as 16- and 8-bit transfers. EISA retains ISA's slow 1 MB/sec DMA for reasons of compatibility, but adds two new ISA-type DMA modes. DMA types A and B increase the throughput of standard ISA DMA for 16-bit cards simply by cutting the number of bus cycles needed per transfer. Most modern expansion cards (including ISA expansion cards) can handle these faster transfer rates. All you need to do to use the new modes is to select one of the new DMA options in the BIOS Setup.

DMA Type C is for EISA boards only. It is also referred to as Burst Mode DMA. As with processor-controlled burst mode, only one bus cycle per transfer is needed, and the maximum burst size is 1 KB. The maximum throughput is 33 KB/sec.

The EISA bus controller - actually called the Integrated System Peripheral (ISP) - implements a sophisticated three-level bus arbitration mechanism which ensures that all devices have a chance to access the bus. Each arbitration cycle has three phases: the refresh phase, the CPU/bus master phase and the DMA phase.

PCSA

The Author

Julian Moss is a PC consultant and writer. He can be contacted via pcsa@oakworth.demon.co.uk.
The concluding half of this article will appear in a future edition of PCSA.

Understanding Windows Security

Mike Lewis offers some tips on how to enhance the security of a Windows-based PC.

This article describes several easy-to-implement techniques for enhancing the security of your Windows-based PCs. They will not stand up to attack by a determined intruder - and perhaps not even by a knowledgeable end-user - but they will go a long way towards protecting your systems from casual meddlers. As well as being easy to set up, they have the advantage of requiring little or no cash outlay. In fact, you probably already own all the tools you need to put these measures in place.

The techniques which I'll describe work with Microsoft Windows 3.1 and Windows for Workgroups 3.11, both on stand-alone and networked PCs. They are not applicable to the forthcoming Windows 4.0 (also known as Chicago), which will have its own centrally-managed end-user security system.

Password Protection

One of the most common security problems faced by end-users is the need to "lock" their systems when they leave their desks for a few minutes. If you are called into a meeting at short notice, you don't want to have to close your application and log out of the network while you are away. Equally, you don't want to leave your machine in a state which permits any nosey passer-by to pry into your work or, as has been known to happen, send abusive email from your account to important people.

The solution to this problem is simple. It relies on that most familiar of Windows software tools: the screen saver. All Microsoft-supplied screen savers have a password-protection feature. With the feature enabled, no-one can break out of the saver - and into the system - without first entering the password (see Figure 1). Although it is a simple technique, it is surprisingly robust. It also has the advantage that end users can manage it for themselves, simply by visiting the Desktop dialogue within Control Panel.

There are, however, two drawbacks. First, there is nothing to prevent an intruder from re-booting the PC and thereby circumventing the password. This is really only a problem on stand-alone machines. If the computer is connected to a LAN, the intruder would have to log onto the network in order to regain access to the system, at which point the network's own password control would come into play.

The second drawback is one of timing. The screen saver kicks in if there is no keyboard or mouse activity after a pre-determined period. This is typically between two and four minutes, which is plenty of time for a co-worker to start messing around with your computer if you are called away from your desk.

You can overcome both these difficulties by running the screen saver as a stand-alone program. To do so, you must first change the extension of the saver from SCR to EXE. Alternatively, you can add SCR to the list of extensions which Windows recognises as belonging to executable files. You do this by editing the Programs= entry in the [Windows] section of Win.Ini, so that it looks something like this:

```
Programs = com exe bat pif scr
```

You can now set up a normal Program Manager icon for the screen saver, with the name of the saver's executable file in the program item's command line. You also need to add "/S" to the command line. This switch specifies that the saver is to be run straight away. Without it, the command would invoke the saver's setup dialogue instead of the saver itself. If you were using the Flying Windows screen saver, for example, the command line might look like this:

```
c:\windows\ssflywin.scr /s
```

You can now start the saver whenever you like, simply by double-clicking the icon.

To prevent an intruder from gaining access to the system by re-booting the machine, place a copy of the icon in the Program Manager Startup group. That way, you will be prompted for the password every time you start Windows. Bear in mind that this does not offer a high degree of security - a knowledgeable intruder could easily by-pass the Startup group and disable the screen saver - but it should at least discourage opportunist snoopers.

Lost Passwords

One problem with any password-based system is that users occasionally forget their passwords - and then make frantic calls to the support staff to rescue them. In the case of Windows screen savers, the rescue operation is a fairly painless one.

The first step is to open the user's copy of CONTROL.INI in a plain-text editor such as Notepad. Scroll down until you reach the [ScreenSaver] section, and look for an entry similar to the following:

```
Password=@mA(1AQ8
```

The characters to the right of the equals sign form an encrypted version

of the password. Delete this line.

Next, look for the section in CONTROL.INI that corresponds to the specific screen saver. For example, if you are using Flying Windows, the relevant section would be [Screen Saver.Flying Windows]. This section contains the following entry:

PWProtected=1

Change the 1 in this entry to 0. If the user has password-protected more than one screen saver, repeat this step for each of them.

When you save CONTROL.INI and restart Windows, the screen saver will still be active but its password-protection will have been removed. The user can now go ahead and set a new password.

Boot-Level Security

If you feel that the screen saver does not offer enough protection against an intruder who re-boots the system, you should consider implementing password control at the boot level. There are many hardware products - all of them expensive - which will help you to achieve this, but there are a number of simple, low-cost software solutions too (see the text box for contact details).

One of the most simple of these programs is a shareware utility called FSK Security. It works by establishing two passwords. The System password is required to boot the PC, while the Config password allows a support person to alter the System password. The program's main weakness is that it is run from the AUTOEXEC.BAT file. It does not take much imagination to find ways of circumventing it - booting from a floppy disk is the most obvious.

A somewhat more secure option is the Safety Boot program, which is also shareware. Like FSK, it provides separate passwords for the user and the support person, although a user who knows the support person's password can use either of them to boot the machine. Unlike FSK, it cannot be by-passed by booting from a floppy. The program works by locking an entire hard drive, so even if you do boot from a floppy, Drive C: will still be inaccessible.

ProgMan Restrictions

As well as preventing intruders from messing around with other people's PCs, you might also want to stop users from fiddling around with their own systems. You might, for example, want to prevent users from creating Program Manager icons or reconfiguring their Program Manager groups. Or you might want to restrict their ability to run applications that they do not have a specific need to use.

You can achieve these aims by specifying restriction levels within Program Manager's INI file. To do so, open the user's copy of PROGMAN.INI in Notepad or another plain-text editor. Scroll to the end of the file, then add a new section, with the header [Restrictions]. You can now add entries to this section to establish the required restriction levels (see Figure 2).

To prevent the user from running programs other than those for which a Program Manager icon exists, add this entry:

NoRun=1

This will cause the Run command on the File menu to be greyed out (remember that you must restart Windows for these INI changes to take effect). If you also want to prevent the user from adding, moving, copying or deleting program items and groups, add this entry:

NoFileMenu=1

As well as removing the entire File menu from Program Manager, this entry will also prevent users from dragging program icons around.

You can also specify restrictions by means of the EditLevel entry. The general form of this entry is EditLevel=n, where n represents the required edit level.

An edit level of 0 means that the user is free to make changes to Program Manager groups and items (this is the default case). If you set the level to 1, users will not be able to create, delete or rename groups, but they will still be able to make changes to individual program items. Setting the level to 2 prevents them from changing either groups or program items.

With an edit level of 3, all the restrictions in edit level 2 apply, and users are also prevented from editing the command line in the program item properties dialogue. Finally, an edit level of 4 means that, in addition to the restrictions in edit level 3, none of the other fields in the group or program item properties dialogues can be edited.

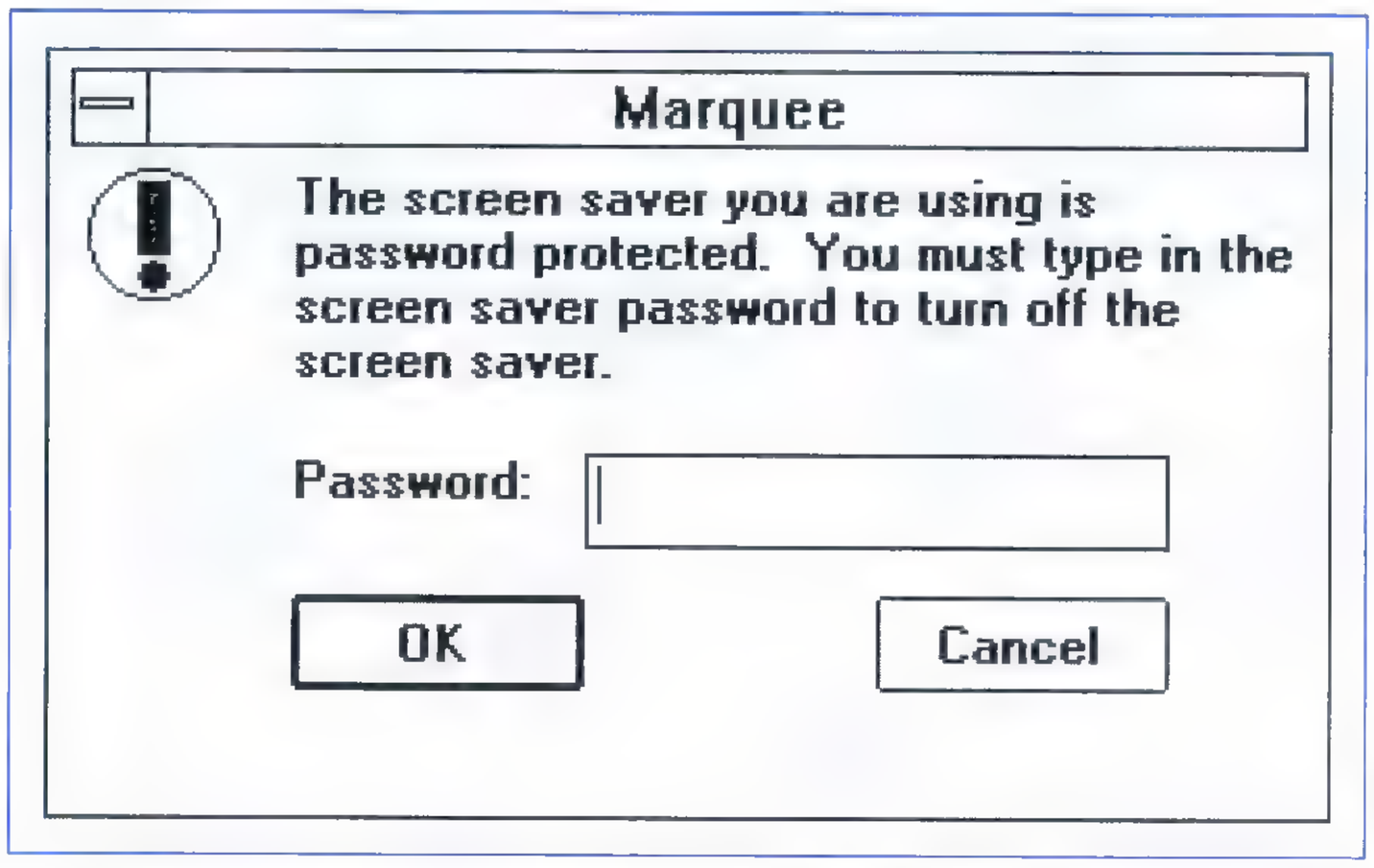

Figure 1 - Screen savers offer simple but effective password control

Windows Security

"If the screen saver does not offer enough protection against an intruder who re-boots the system, you should consider implementing password control at the boot level. There are many hardware products which will help you to achieve this, but there are a number of simple, low-cost software solutions too."

There are two further entries that can be added to the [Restrictions] section. NoClose=1 prevents the user from exiting Windows from Program Manager. NoSaveSettings=1 disables the Save Settings on Exit toggle in the Options menu. With this entry in force, any changes which the user makes to the arrangement of windows and icons will not be saved when the user exits Windows, regardless of the previous setting of this toggle.

One problem with all this is that users can easily remove the restrictions simply by editing PROGMAN.INI themselves. You can go some way towards discouraging this by flagging the file as read-only, or, better still, as hidden (use either File Manager's File Properties command or the DOS ATTRIB command to set these attributes). When the user starts Windows, a message will appear warning that any Program Manager changes will not be saved, but this does prevent the user from making changes (for example, to the size and position of the overall Program Manager window) during the current session.

Another difficulty is that users will still be able to use File Manager to run *ad hoc* programs, regardless of the restrictions enforced within Program Manager. They will also be able to drag and drop files from File Manager to Program Manager, even if there are Program Manager restrictions in force which prevent them from creating new program items.

The only way round this is to remove the File Manager icon from Program Manager, thus preventing the user from running File Manager at all. This is not necessarily as draconian as it sounds, given that many major Windows applications have file management features of their own. And if you are creating the sort of tightly controlled environment which these Program Manager restrictions imply, you would probably want to restrict access to file management functions anyway.

You might even go one step further by doing away with Program Manager entirely. If the user only needs to run one main application - the company's standard word processor or spreadsheet, for example - you could specify that application as the Windows shell. When the user starts Windows, the application would be the only program that he or she sees; when the user quits the application, Windows would close down too.

To achieve this, edit the Shell= entry in the [boot] section of System.Ini. Remove the reference to PROGMAN.EXE in this entry, inserting instead the command which starts the relevant application. For example: Shell=C:\WINWORD\WINWORD.EXE. This establishes Word for Windows as the shell.

File Encryption

The final technique is aimed at stopping snoopers from peering into other people's data files. Many Windows applications allow users to password-protect individual documents, though the degree of protection which this provides varies widely. In Word for Windows, for example, assigning a password to a file will

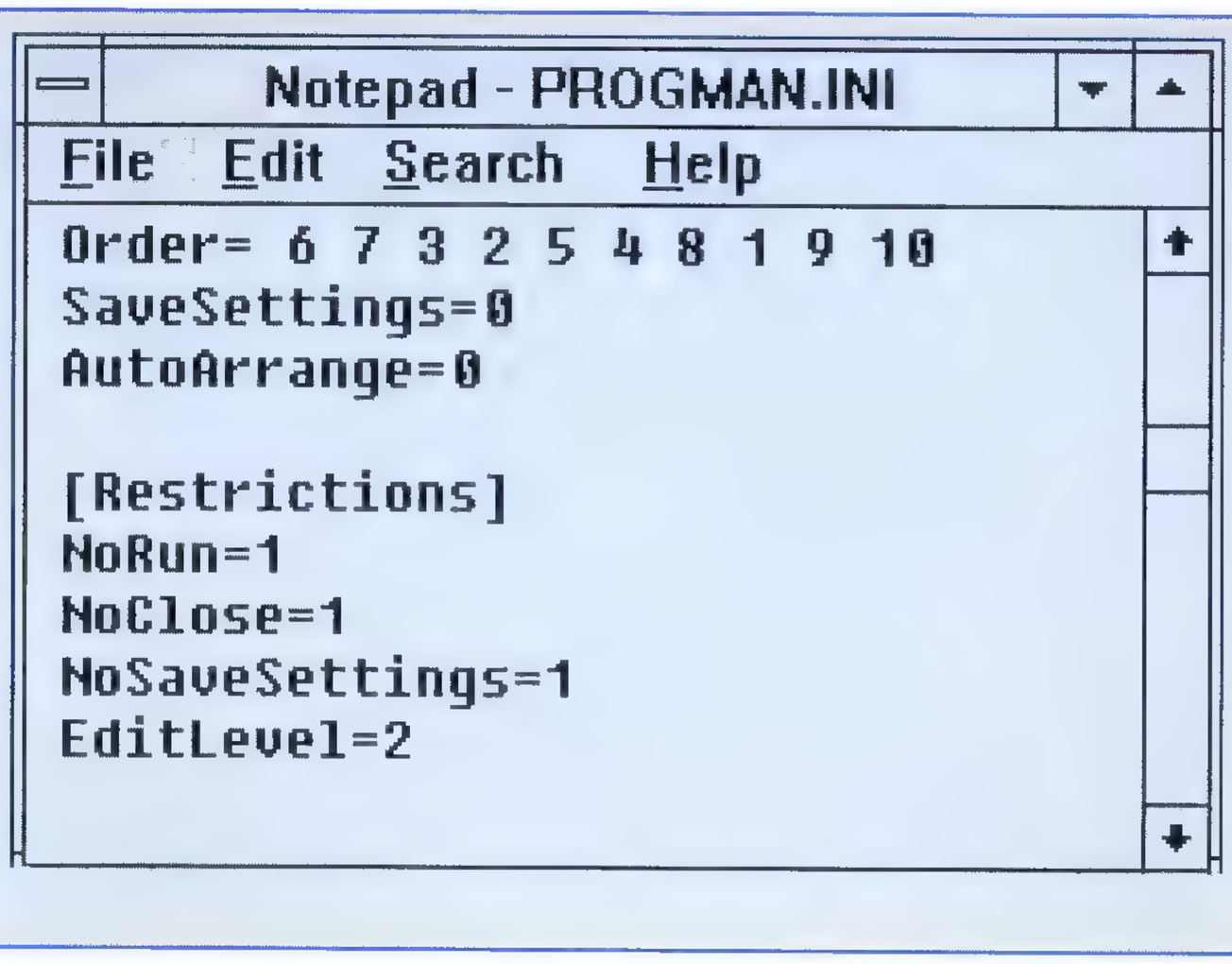

Figure 2 - PROGMAN.INI restrictions prevent users messing around with Program Manager

prevent unauthorised users from opening the document within Word, but you don't have to be a genius hacker to find ways of circumventing it.

A better solution is to encrypt the file. File encryption is not really suitable for live data files, as it would require you to decrypt the file every time you access it and to re-encrypt it after each edit. However, it can be a valuable security aid when you pass files around on floppy disks, attach them to mail messages, or place them in long-term storage on a file server.

The most common file encryption program is one that you probably already own. It is PKWare's famous utility, PKZIP. Although mainly used for file compression, this handy program provides a high standard of file encryption as well. To use it, you add the -s switch to the PKZIP command line, immediately followed by your choice of encryption key.

For example, the following command encrypts all the Word documents in the C:\LETTERS directory, using the key COFFEE. The encrypted files (which will also be compressed) will be stored in a new file named MYDATA.ZIP:

```
pkzip mydata.zip -sCOFFEE c:\letters\*.doc
```

The key is case sensitive and may be as long as you like, subject to the restrictions imposed by the DOS command line. If you want it to include spaces, place the entire key in quotes.

If you don't like the idea of having the key appear on the screen as you type it, use the -s switch without the key. The program will then prompt for the key, which it echoes as asterisks. In this case, the key is limited to 64 characters, but it may include characters which are not normally permitted in a DOS command line - the Tab character, for example.

You can also encrypt individual files when you add them to an existing ZIP file. It does not matter if the ZIP already contains encrypted files, or if those files are encrypted with a different key. In fact, a given ZIP may contain any combination of encrypted and unencrypted files, with the same or different keys.

To decrypt the file, run PKUNZIP, using the -s switch in the same way as with PKZIP. For example:

```
pkunzip mydata.zip -sCOFFEE
```

This will extract all the files from MYDATA.ZIP which are either unencrypted or whose encryption key is COFFEE, with the encrypted files being decrypted in the process. Any files which were encrypted with a different key will be skipped (a message is displayed to warn you of this - see Figure 3.

Conclusion

The techniques described here are all safe, inexpensive and easy to implement. On their own, they will not provide a defence against a determined attacker. But, if you use them in conjunction with other sensible security measures, you will find that they can do a great deal to improve the overall security of your installation.

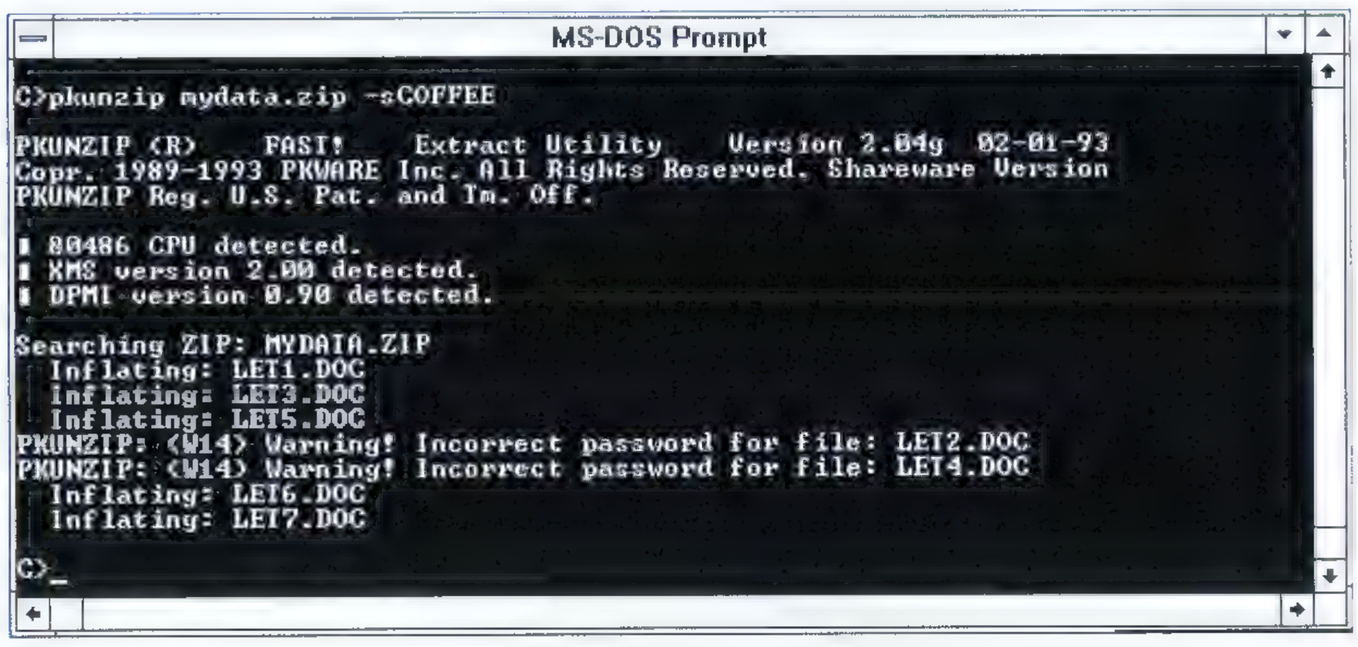

Figure 3 - PKWare provides all the file encryption features you are likely to need

Products mentioned

All third-party products mentioned in the article are available as shareware. The contact details are as follows:

FSK Security: Vladimir Kilin, 7323 17 Avenue, Brooklyn, NY 11204-5152, USA. Registration: US$5.00.

PKZIP: PKWare Inc., 9025 N. Deerwood Drive, Brown Deer, WI 53223-2437, USA. Registration: US$47.00.

Safety Boot: Cam Belgium, Chaussee de Tubize, 483 A 1420 Braine-l'Alleud, Belgium. Fax: +32 (2) 387.28.90. Registration: US$39.95.

PCSA

The Author

Mike Lewis is a freelance technical journalist. He writes for computer journals throughout the world, and has written and translated several computer-related books. You can contact him on CompuServe (100012,2105), or at pcsa@oakworth.demon.co.uk.

Supporting OS/2

Tim Sipples presents a selection of the most frequently asked questions about IBM OS/2.

I am having trouble installing OS/2 2.1. Are there any tips and tricks I should know?

Midway through installation of OS/2 2.1 from CD-ROM, do not place a check mark in the box next to the CD-ROM Device Support option on the System Configuration screen. Do not attempt to view or change this CD-ROM drive selection.

When installing OS/2 2.1 from a CD-ROM drive that requires a driver not supplied by IBM, copy the OS/2 .ADD driver file to a copy of Diskette 1 and add the line:

BASEDEV=xxxxxxxx.ADD

to the end of the diskette's CONFIG.SYS file. After installation, if your CD-ROM drive is not functioning correctly, follow these steps (changing "C:" if necessary):

1. Copy the files OS2CDROM.DMD and CDFS.IFS from Diskette 1 to directory C:\OS2.
2. Edit CONFIG.SYS and add the following lines to the end of the file:
   BASEDEV=OS2CDROM.DMD /Q
   IFS=CDFS.IFS /Q
3. Reboot.
4. Start Selective Install from the System Setup folder. Check the CD-ROM Device Support checkbox on the System Configuration screen. Click OK to display the list of CD-ROM drives. Select the appropriate choice (usually OTHER) and click OK.
5. Continue the Selective Install procedure until complete, then reboot.

If you have formatted any of your hard drives with HPFS under OS/2 2.0, and you have upgraded to OS/2 2.1, go to an OS/2 command line (window or full screen) and type:

CHKDSK x:

where x is the drive letter. Repeat for all HPFS drives on your system. If you see the error SYS0551 you should immediately contact IBM for the "OS2DASD Fix," usually filename 21DISK.ZIP.

Some PCs have trouble printing under OS/2 2.1. This problem can often be traced to an interrupt conflict, a substandard cable, an interfering software security "dongle" or a faulty printer adapter. LPT1 uses IRQ 7, and LPT2, if installed, uses IRQ 5. Interrupts should not be shared on AT bus machines. The SoundBlaster, for example, comes set to IRQ 7. Reset it to an unused interrupt.

Make sure adapters with onboard ROMs are not conflicting with other adapters. For example, many SuperVGA adapters use large segments of upper memory, and many hard disk adapters have onboard ROMs which can be mapped into the same areas. Adapters must not share address space or interrupts. Check your product manuals for more help. To resolve conflicts, try writing down the DMA channel(s), interrupt (IRQ) level(s), I/O or port addresses, and ROM or upper memory addresses used by every device in your system. Conflicts should then be readily apparent.

Be sure adequate free disk space is available before installing, including space for a swap file. Drives compressed using Stacker, DoubleSpace, or similar utilities should be uncompressed before installing (unless access to these drives from OS/2 is not needed.

Do not select HPFS when installing if your machine has 6 MB of RAM or less, or diminished performance will likely result. Change the IFS=...HPFS line in your CONFIG.SYS to REM IFS=... if necessary.

Be sure your CMOS setup parameters are set correctly, especially those relating to floppy drives. RAM should be given sufficient wait states and precharge cycles. Test with cache memory and/or shadow RAM disabled if necessary. The AT bus should run at 8 MHz. For best performance, make sure all your RAM is set to be cacheable.

If you are using the IBMINT13.I13 driver to access an MFM, RLL, or ESDI hard drive, and the hard drive has more than 1024 cylinders, be certain your hard disk adapter's sector translation mode is enabled. Regardless of the driver you use, make sure your boot partition is completely inside the first 1024 cylinders.

Older Adaptec SCSI adapters may require a free BIOS upgrade to recognize hard disks larger than 1 GB. Some Quantum LPS105AT IDE hard disks require a free ROM update (to Version 2.6) from the manufacturer to work with OS/2.

If your AMI keyboard BIOS is below Revision F you may need an update. Contact Washburn & Associates (phone: +1 716 248 3627) for an inexpensive replacement.

Check to make sure keyboard DIP switches are set correctly. For example, if the keyboard is attached to a system with an AT bus it should typically be switched to "AT" mode.

"Autoswitching" on non-IBM EGA adapters should be disabled (usually with a DIP switch or jumper setting). In rare cases it may be necessary to switch third-party VGA/SuperVGA adapters into 8-bit mode and/or disable "autosense".

OS/2 is particularly sensitive to bad RAM or cache memory (often reflected in TRAP 0002 error messages). Use a thorough RAM testing utility,

and try not to mix 9-chip and 3-chip SIMM/SIPP memory modules. When upgrading, avoid adding RAM which is not rated (in nanoseconds) at least as fast (ie, with an equal or lower number) as the RAM already in the system.

Allow several minutes for OS/2 2.1 to build your desktop (and display icons) at the end of installation - take the Tutorial offered to you in the meantime. Avail yourself of the "Start Here" icon, the other online help, and the README file located in the root directory. They will help in getting started with the Workplace Shell and in properly configuring your system.

When installing over a beta version of OS/2 be sure to reformat.

A Guide To IBM OS/2-speak

APAR
A bug fix which has been (or will be) created by IBM to address a very specific problem.

CID
Configuration/Installation/Distribution. A term usually used to refer to the ability to install an operating system or application remotely, over a network. (Example: "IBM TCP/IP 2.0 for OS/2 is now CID-enabled.")

CSD
Corrective Service Diskette(s).

DASD
Direct Access Storage Device. IBM-speak for a hard disk, and pronounced "dazzdy".

DMA
Direct Memory Access. Circuitry provided on all PCs to allow peripherals (such as disk controllers) to transfer data to memory directly, without the assistance of the computer's processor. Appropriate use of DMA can often help to improve overall system performance.

EA
Extended Attribute. Up to 64 KB of assorted data stored with any file under OS/2. Such data may include file type (eg, Plain Text), icons, comments, and other information which is best left outside the file itself.
Only OS/2 applications can create and modify extended attributes.

ES
Extended Services.

FAT
File Allocation Table. The disk format introduced by DOS.

GA
General Availability. Available for purchase as a shrinkwrapped product from IBM and its dealers.

HPFS
High Performance File System. See the main text.

IFS
Installable File System. Refers to an OS/2 driver used to manage a file system type. Available IFSes include NFS (used with TCP/IP networks), CD-ROM, HPFS, and HPFS386 (supplied with IBM LAN Server Advanced).

IPL
Initial Program Load. Starting a PC's operating system (ie, booting or rebooting).

LA
Limited Availability. Available only from IBM to certain customers.

To install the Upgrade Edition of OS/2 2.1, DOS or OS/2 must already reside on the hard disk. If Diskette 1 is not write protected then the installation program will record a file indicating that upgrade terms have been satisfied and, in the future, will not require DOS or OS/2 on the hard disk to install.

OS/2 uses the same diskette format as DOS, so use DISKCOPY to backup the installation diskettes and verify that none has been corrupted.

Owners of IBM PS/2s should make sure that any applicable ECAs (engineering changes) have been performed and that the most recent Reference Diskette is in use. Reference Diskettes are available from the IBM BBS on +1 919 517 0001.

Try disconnecting any tape backup device if "Cannot find COUNTRY.SYS" messages are encountered when booting OS/2. This error message may also indicate that OS/2 cannot find its boot drive, usually because of an improperly loaded or omitted hard disk .ADD device driver.

An Always IN-2000 SCSI adapter with BIOS 3.06A or 3.20 requires an updated version from the manufacturer. A companion 8-pin serial PROM chip may also need to be updated. Contact Always at +1 818 597 9595. Also ask about non-IBMINT13.I13 driver support.

The TI TM4000 notebook may require a BIOS update to run OS/2 2.1; phone +1 817 771 5856 for help. Also ask about an OS/2 driver for the QuickPort trackball. Before installing OS/2 on this machine, be sure to set Advanced OS Power off and HDD Motor Timeout Always On in the second page of the machine's setup screen.

I can't install OS/2 2.1 from Drive B:. What's wrong?

IBM OS/2 2.1 can only be installed starting from Drive A:, like DOS (unless your BIOS supports booting from Drive B:). After booting from Drive A:, OS/2 can then be copied from CD-ROM or across a network. (For more information on installation across a network, see Remote Installation and Maintenance, IBM Publication No. GG24-3780.)

Supporting OS/2

If you have the wrong disk size go back to your dealer and obtain the correct media. Otherwise you could open your machine and swap floppy drive cable connectors, use your system's setup utility to set the new CMOS parameters, and then install OS/2 from the "new" Drive A:. Sometimes the floppy drive cable connectors will not be the same. If so you can obtain an adapter plug.

How do I access HPFS partitions on my hard drive without booting from the hard drive? I'm getting error messages - how do I "repair" my hard disk?

With IBM OS/2 2.1, insert the Installation Diskette, Shutdown (if necessary), and reboot. When prompted insert Diskette 1 and press ENTER. When prompted, press ESC. You will be given an OS/2 command line prompt. From there you can make necessary changes to your hard disk - an OS/2 character mode text editor on diskette is handy for such changes. (Make sure you backup CONFIG.SYS before making any changes so that you can easily revert to the old version should things go wrong.)

You may use this diskette boot method to run CHKDSK on your FAT or HPFS volumes. After you reach the command line, insert Diskette 2. Do not log to another drive. Type CHKDSK X: /F to repair most kinds of damage to your hard disk, replacing X with the appropriate drive letter. OS/2 CHKDSK will also mark your hard disk as accessible, if possible, should OS/2 "lock it out" for some reason. It will also allow Workplace Shell drive objects to open properly if they are not functioning correctly. Repeat twice for each drive letter you wish to check and/or repair.

"Errors" may be reported by CHKDSK if OS/2 was booted from the hard disk. These "errors" are normal. Since the hard drive is in use by OS/2 itself (and files are open) CHKDSK is unable to accurately report errors.

The best way to avoid the need to perform CHKDSK is to always select Refresh then Shutdown. Click on the Workplace Shell desktop background using mouse button two to bring up the appropriate menu. Also, avoid manipulating OS/2-related files when using native DOS. Finally, enable autochecking for all your hard disk volumes. For HPFS volumes use the /AUTOCHECK parameter in the IFS=...HPFS line in your CONFIG.SYS. For FAT volumes use the AC parameter in the DISKCACHE line of your CONFIG.SYS. See the online Command Reference for details.

What are CSDs, how do I tell which I have, and where do I get them?

CSDs are Corrective Service Diskettes, or bug fixes (Service Paks), periodically issued by IBM. The OS/2 CSD level number may be obtained using the command SYSLEVEL from an OS/2 command line prompt. CSDs are cumulative, ie, only the most recent CSD is required to bring a system up from any previous CSD level. However, CSDs only apply within a major version number. For example, an upgrade, not a CSD, would bring OS/2 Version 2.0 up to Version 2.1. Note also that each national language (eg, French, UK English) uses a distinct CSD.

Your local IBM office will tell you how to order CSDs.

A Guide To IBM OS/2-speak (Continued)

Multitasking
Running two or more applications "simultaneously," dividing the computer processor's attention among them. (In fact, the two or more applications only appear to run simultaneously because the processor switches between them rapidly.)
Cooperative multitasking, such as that found in Microsoft Windows and Macintosh System 7, requires that each application be written so as to "surrender" the computer's processor at regular intervals so that it can devote attention to other running applications. If one application for some reason refuses to yield the processor, all other applications stop running. Preemptive multitasking, as found in OS/2 and UNIX, for example, leaves the operating system in charge of delegating processor time to each running application. The amount of attention given depends on the operating system's scheduler, the logic which assesses (and perhaps adjusts) the priorities of various tasks and assigns processor attention accordingly.

Multithreading
An operating system's ability to manage what are sometimes called lightweight processes, namely subtasks which are spawned by applications. For example, a word processor may be written so that any printing operation is put in a separate thread. This thread is then run alongside the word processor itself, in the background, so that control returns immediately to the user of the word processor.

PM
Presentation Manager. The underlying services used by programmers and the Workplace Shell (see WPS) to provide windows, scroll bars, dialog boxes, and other essential interface elements.

PMR
Problem Management Record. A number assigned by IBM to track a customer-reported problem.

RIPL
Remote Initial Program Load. The capability to boot (start) a PC (load its operating system) over a network. See IPL.

Seamless
Refers to the ability to run Windows applications alongside OS/2 and DOS applications on the Workplace Shell (see WPS) desktop as opposed to the full screen Win-OS/2 desktop.

I normally use UNIX. How can I make OS/2 resemble UNIX?

A great number of GNU and UNIX utilities have been ported to OS/2 native mode and are available from various shareware and freeware sources. A uucp package, UUPC/Extended, is available via anonymous ftp from ftp.clarkson.edu, directory /pub/uupc; mail help@kew.com with questions.

In addition, the Hamilton C Shell is available from Hamilton Labs (phone: +1 508 358 5715 or mail 3890321@mcimail.com). The Thompson Toolkit, a Bourne-like shell, and awk are published by Thompson Automation (phone: +1 206 224 1639). MKS (phone: +1 519 884 2251 or mail pat@mks.com) publishes a number of standard UNIX utilities for OS/2. Hippix (Hippo Software, phone: +1 801 531 1004) provides a set of low cost UNIX-like command utilities (such as grep, awk, sh, and vi) along with a POSIX programming library. An OS/2 version of lint is available from Gimpel Software (phone: +1 215 584 4261). For OS/2-specific X Windows server support, IBM provides an optional package available with its TCP/IP 2.0 for OS/2.

How do I start a background process from the OS/2 command line?

Look up the START and DETACH commands in the online Command Reference. If you wish to start a DOS session with non-default settings, use a utility such as STARTD. If you wish to start an OS/2 session from a DOS session, try OS2EXEC. Both (and several others) are available from various shareware and freeware sources.

How do I add new Adobe Type Manager typefaces?

OS/2 2.1 comes with built-in Adobe Type Manager (ATM) for OS/2 and Win-OS/2. A basic set of typefaces (Courier, Helvetica, and Times New Roman) comes with OS/2 2.1 and is installed (if selected) for use under both OS/2's and Win-OS/2's ATM.

Each typeface you install under OS/2 and/or Win-OS/2 should come with at least two separate files with PFB and AFM extensions. To install a typeface for use under Win-OS/2, use the ATM Control Panel. The Win-OS/2 ATM Control Panel will then build a PFM file from the AFM file if a PFM file is not already included. To install a typeface for use with OS/2-specific applications, select OS/2 System -> System Setup -> Font Palette -> Edit Font -> Add.

How do I tweak OS/2 2.1 for maximum performance?

For OS/2 overall, the CONFIG.SYS parameters MAXWAIT, TIMESLICE, PRIORITY, PRIORITY_DISK_IO, PROTECTONLY, and cache settings (in the DISKCACHE line, for FAT; or IFS line, for HPFS) can be tweaked. The swap file should be placed on the most used partition on the least used hard disk, and its location is controlled by the SWAPPATH line. See the online Command Reference for details.

FAT partitions should be periodically defragmented. A shareware defragmenter for DOS called DOG (Disk OrGanizer) works well, as do many others. (You can boot DOS from a floppy disk to run such a utility.)

For the Workplace Shell, drag shadows of most often used items to the desktop or to folders closer to the "surface" - opening folders takes time. Drag shadows of program objects you use often (eg, the Win-OS/2 full screen Program Manager) to the Startup folder. Disable animation (go to OS/2 System -> System Setup -> System -> Window). Use the faster Details View when opening drive and folder objects; to set Details View as the default, open the settings notebook for the object, select the Menu tab, click on ~Open, then the Settings button, then select the Default Action.

Try reducing the number of on-screen colours or dropping down in screen resolution to enhance speed. Close (not just minimize; check the Window List) unnecessary objects and applications. Use the Monochrome scheme from the Scheme Palette - it provides marginally faster screen updates. Consider adding more RAM.

For DOS programs, run full screen instead of windowed if speed is important. In DOS Settings for each application: reduce conventional, XMS, DPMI, and EMS memory allocations to the bare minima required for maximum performance; turn off VIDEO_RETRACE_EMULATION

A Guide To IBM OS/2-speak (Continued)

SMP
Symmetric Multiprocessing. A set of technologies in which two or more computer processors (CPUs) are managed by one operating system to provide greater computing power to applications. With SMP, processors are treated more or less equally (with applications able to run on any or perhaps all processors in the system, interchangeably, at the operating system's discretion). Simple MP usually involves assigning each processor to a fixed task (such as managing the file system), reserving the single main CPU for general tasks. OS/2 currently supports so-called HMP (Hybrid Multiprocessing), a version of MP which provides some elements of SMP, using add-on IBM software called MP/2. OS/2 SMP is slated for release later this year.

SP
Service Pak. See CSD. Sometimes numbered (eg, "SP 2") to refer to a particular Service Pak.

Win-OS/2
IBM's customized version of Windows, based on Microsoft's own source code, which provides compatibility with Windows applications under OS/2.

WPS
Workplace Shell. OS/2's most commonly used user interface which provides icons, folders, drag-and-drop configuration, settings notebooks, and other features necessary for user interaction with the operating system and its applications.

Supporting OS/2

unless necessary; adjust IDLE_SENSITIVITY; turn off DOS_BACKGROUND_EXECUTION if not needed; change the HW_TIMER setting (particularly for games); enable VIDEO_FASTPASTE if possible; turn on HW_ROM_TO_RAM. Communications programs should use hardware handshaking where possible (use OS/2's MODE COMx command if necessary), and a buffered UART can prove helpful. (DOS programs running under OS/2 will not be aware of a buffered 16550AF UART. OS/2 virtualizes the serial port and manages the buffer itself.) For faster printing set the DOS program's output port to LPTx.OS2 (where x is the printer port number) - use a "print to file" option if necessary. Disable any DOS print spoolers; rely on OS/2's spooler instead. Increase CONFIG.SYS's PRINTMONBUFSIZE values. Other, standard steps to enhance DOS performance (eg, increasing BUFFERS in CONFIG.SYS) of course apply.

For Windows programs, run using a full screen desktop if speed is vital. The Win-OS/2 Full Screen icon set up by the installation program has poor Settings. For better performance perform some of the same steps outlined in the preceding paragraph, including turning VIDEO_RETRACE_EMULATION off. The same printer output advice also applies. Consider disabling the Public setting in the Clipboard. If available, set VIDEO_8514A_XGA_IOTRAP to off. If mouse control is lost when switching to/from the Win-OS/2 session, try setting VIDEO_SWITCH_NOTIFICATION off.

How do I measure OS/2 performance and memory usage?

OS/2 does not treat system resources like DOS. Memory is treated as a virtual resource, used intelligently. For example, OS/2 will retain unused, "dormant" code in memory if that memory is not otherwise required, on the assumption that that code may be used again. Also, all but a small portion of OS/2 (and most applications, no matter how many are running) may be paged to disk should a large amount of physical memory be required. Utilities which display "free" memory, then, are only useful for rough, relative measurements. (Such utilities also often fail for another reason: many only report the largest contiguous block of free physical RAM. And a few will never report more than 16 MB of RAM because they were designed for OS/2 1.x.)

Similarly, utilities which purport to measure system load (eg, Pulse) should not be relied upon for definitive performance measurement. Subjective assessments are often much more reliable. Pulse (and similar utilities) rely on a measurement of processor time allocated to a thread running at OS/2's lowest priority. This method is sometimes subject to erroneous results.

That said, more rigorous system performance optimization and monitoring tools include System Performance Monitor/2 (IBM Program No. 5871-3415), BenchTech (Synetik, phone: +1 303 241 1718), OR/SysMon (International OS/2 User Group based in the UK, phone: +44 285 641175), CPU Monitor (Bon Ami), and Performance 2.1 (Clear & Simple, phone: +1 203 658 1204).

Note that OS/2's swap file is designed to behave with hysteresis. It will not shrink in size as easily as it grows, under the assumption that swap space needed once may be needed again. It should shrink given enough time and continued, less intense system loads.

My background bitmap does not display correctly. What's wrong?

Colour bitmap images used for the Workplace Shell screen or folder backgrounds may not display correctly (they may have distorted or missing colours) due to incorrect matching with OS/2's default palette. Unlike Windows, OS/2 does not adjust the palette to accommodate background bitmaps (to keep the rest of the desktop from experiencing colour distortions). (Palette control is now available to applications running under the 32-bit graphics engine with an appropriate display driver, however.)

To remedy the problem you may use the numerous background images which have been specifically prepared for the Workplace Shell, available from shareware libraries.

Or, use an image editing/conversion utility which can create a proper, system palette-matched bitmap file. For example, JoeView (shareware) may be used to import noninterlaced GIF, Windows BMP, and PCX files and save them as palette-matched OS/2 BMP files.

Note that background bitmap images impose some additional overhead, taking up RAM and disk resources. You should probably use them sparingly. Also, if you have set a Win-OS/2 background bitmap you may experience desktop colour distortions when running Windows programs "seamlessly." Disable the Win-OS/2 background bitmap to remedy the problem.

How do I boot a real version of DOS from within OS/2 2.1?

Booting a real version of DOS under OS/2 provides certain features that the OS/2 emulated DOS sessions cannot. For example, a specific DOS session can provide access to devices (like CD-ROM drives) and networks for which there are only DOS device drivers. A specific DOS session can also help get DOS applications which generate spurious "divide by zero" errors running again.

You will be able to run one such session per hardware device. So, for example, if you have your DOS networking software loaded in one specific DOS session, you may not start another, similar session.

Specific DOS sessions are discussed in the online Command Reference (under VMDISK), the Master Help Index, and the printed Installation Guide (Appendix E). You should consult those resources first. However, if you are still unsure how to configure your system to run specific DOS sessions, follow these steps:

1. Create a bootable DOS diskette. Insert your DOS system diskette into Drive A: and reboot. When you arrive at the "A>" prompt,

type FORMAT A: /S and press ENTER. (Note that you may wish to format the diskette for the smallest capacity possible, to save hard disk space later on. For example, a 5.25 inch double density - not high density - diskette may be formatted to just 160 KB by adding the /1 /N:8 parameters to the FORMAT command.) When prompted, insert a blank diskette into Drive A and press ENTER. When the FORMAT operation is complete, remove the diskette and restart OS/2.

2. Copy FSFILTER.SYS to the diskette. Double click on OS/2 System -> Command Prompts -> OS/2 Window. Insert the diskette you just formatted into Drive A:. Copy the following file to your startable diskette: \OS2\MDOS\FSFILTER.SYS.
3. Set up CONFIG.SYS. Using a text editor (like the OS/2 System Editor) create the file A:\CONFIG.SYS with the following lines at the top:
   DEVICE=FSFILTER.SYS
   DEVICE=C:\OS2\MDOS\HIMEM.-SYS
   DEVICE=C:\OS2\MDOS\EMM386.SYS
   DEVICE=C:\OS2\MDOS\ANSI.-SYS
   Change the "C:" drive letter if OS/2 is installed on another drive. Add any other lines as required for your application (like CD-ROM or networking), but do not include any XMS, EMS, mouse, or memory management device drivers. Make sure that everything is referenced with a drive letter and path, as above.
4. Set up AUTOEXEC.BAT. Likewise, create a file named A:\-AUTOEXEC.BAT and make sure that the first line reads:
   C:\OS2\MDOS\MOUSE
   changing "C:" if necessary. Add any additional lines (like PATH, SET PROMPT and so on) as required by your application. Make sure that \OS2\MDOS is referenced in the PATH.
5. Test your DOS diskette. Once you have configured the CONFIG.SYS and AUTOEXEC.BAT files as you wish, double click on OS/2 System -> Command Prompts -> DOS from Drive A:. A DOS session should start. Test for the functionality you need (like access to your CD-ROM reader or network). If the session is not working properly, press CTRL-ESC and shut down the session, edit CONFIG.-SYS and/or AUTOEXEC.BAT as required, and repeat the test.
6. Create the diskette image. When you are satisfied that your specific DOS session diskette functions properly, go back to the OS/2 Window and type VMDISK A: C:\DOS.IMG to create a diskette image file. (If you want the file to be located on another drive or in another directory, change "C:\" accordingly.)
7. Create a program object for your specific DOS session. Drag a program object from your Templates folder to any target folder. When the notebook opens, enter a single asterisk (*) in the Program Name field, then click on the right arrow in the lower right. Select either DOS Window or DOS Full Screen for the session type, as desired. Click on the DOS Settings button, and scroll down until you find the DOS_STARTUP_DRIVE property. Enter C:\DOS.IMG in the field at the upper right. (If your image file is not located on Drive C in the root directory, make the necessary changes.) Change any other DOS Settings if necessary. Click on the Save button, then click on the General tab. Give your program object a name. Then close up the notebook.

You should now be able to double click on your new program object to start your specific DOS session. If you require access to your diskette drive (Drive A:), use the FSACCESS command. See the online Command Reference for details.

When formatting your bootable DOS diskette, you may wish to use additional command line parameters to create a diskette with a reduced capacity. The "smaller" the diskette, the less room the diskette image file created by VMDISK will take on your hard disk. See your DOS manual for details, or use the example given above.

How can I create INF files?

Creating INF files is straightforward. All you need is the Information Presentation Facility Compiler (IPFC), part of the IBM Developer's Toolkit for OS/2 2.1 (available separately as IBM Part No. 61G1416 or as part of many development environments such as Borland C++ for OS/2, IBM C Set ++/2 and First Step, and CA-Realizer), and a text editor (like the Enhanced Editor included with OS/2).

Online IPFC documentation is included with the Toolkit, but you may also wish to order the printed Information Presentation Facility Guide and Reference, IBM Publication No. S10G-6262.

If you wish to include illustrations in your INF file you can use any graphics software which can generate OS/2 bitmaps and/or metafiles. (For example, you may create your illustration in PM Chart, paste the illustration into Picture Viewer, then save the illustration as a metafile. Both PM Chart and Picture Viewer are included with OS/2 2.1.) A screen capture utility (like PM Camera or Galleria, available as Shareware) can also prove useful.

PCSA

Acknowledgement

This article is based on the OS/2 Frequently Asked Questions list, compiled and maintained by Timothy F. Sipples, and is used with permission. The author can be contacted via email as usib58c5@ibmmail.com.

How To Troubleshoot Stacker 4.0

We answer some of the most common questions that Stac's technical support staff are currently being asked.

When we run CHECK, we are occasionally receiving warnings from Stacker 4.0 that the two FATs are not identical. What is the recommended way to recover from this?

After issuing the warning, CHECK will advise you that it will run its integrity checks a second time, using the alternate copy of the FAT. Errors may be reported, but usually only on one of the two FATs. After that, you will be presented with an option screen that gives you three choices:

1 - exit so you can try using the first copy of the FAT
2 - exit so you can try using the second copy of the FAT
3 - to copy the first FAT over the second FAT and let CHECK repair any errors

Your goal is to make your current (first) copy of the FAT the error-free version. Then you can use that copy to repair the error. After you receive the warning that the FATs are not identical, do the following:

1. Select option 1. If you ran CHECK from DOS, you will be returned to the DOS prompt. If you started the integrity check from the Stacker Toolbox in Windows, you will be returned to Windows. If so, exit Windows and go to the DOS prompt.
2. Type CHECK.
3. Notice whether the errors on the drive are reported when CHECK is running the first copy of the FAT or the second. If the errors are on the first copy, go to step 4. If they are on the second copy, skip to step 5.
4. Select option 2. Wait until you are returned to the DOS prompt.
5. At the DOS prompt, type CHECK /F.
6. When presented with the options again, select option 3. Let CHECK repair the drive.

How should a user tune Stacker to get the best results in terms of speed and compression ratios?

You must decide whether you wish to use the Stacker Toolbox to tune your computer for maximum speed or maximum compression, based on your system's overall performance and your need for disk space. Stacker provides the flexibility to make use of maximum speed during day-to-day operations, while periodically optimizing your drive to get the best compression. Utilizing this strategy you would use the Stacker Toolbox to select MaxSpeed. But you would also regularly use the Optimize tool to recompress your drive selecting the MaxSpace option.

From Windows:

1. Open the Stacker Toolbox and click on the "tune" button.
2. Select the speed setting you desire. Select MaxSpeed for the fastest performance.
3. When Optimizing your Stacker drive, click on the Optimize button in the Stacker Toolbox.
4. Select the optimization method you desire. To get the best compression select Full-MaxSpace.

From DOS:

1. Type STAC at the DOS prompt.
2. Select Stacker Tuner from the menu.
3. Select the speed setting you desire. "Fastest speed and standard compression" is the equivalent of MaxSpeed in Windows.
4. When Optimizing the drive, select Stacker Optimizer from the main menu.
5. Select the compression method desired. "Full-MaxSpace" gives you the best compression.

Remember that you only need to set the Tuner one time. Your computer will operate at that speed setting from then on, unless you change the setting. You should periodically run the Stacker Optimizer when your fragmentation level gets too high, or you wish to recompress the drive for best compression.

Other Speed Issues

Other things that affect system speed are the settings of your disk cache and setting up a Windows Permanent Swap File.

Most disk caches have the flexibility to change the amount of memory they are devoting to the cache. For example, you can set the amount of memory DOS's SMARTDRV will use in both DOS and Windows. If you are using SMARTDRV, its load line may look something like "C:\WINDOWS\SMARTDRV /X 1024 2048" in the AUTOEXEC.BAT file. The first figure, 1024, indicates the cache size in DOS. The second indicates the cache size in Windows. By experimenting with these parameters you may be able to increase your system's apparent speed. See your disk cache's documentation for details.

Set up a Windows Permanent Swap File using either the 386-Enhanced icon or Virtual Memory icon in the Control Panel. See your Windows documentation for recommended sizes for the WPSF.

How do I make a PC boot disk that can access a hard disk that's compressed with Stacker?

You should have a startup (bootable) disk available to protect data from a system failure. Since you now have drives that are compressed by Stacker, you need to make your startup disk Stacker-aware. When you start your system with this disk, it will start Stacker and provide access to your data.

Here's how to create a Stacker Startup disk if your DOS version preloads compression:

1. Insert a formatted blank disk into your A: drive.
2. At the DOS prompt, type SYS A:. This transfers both the DOS system information and the Stacker information to your disk and makes it Stacker-aware.
3. In order to provide additional speed when booting from this disk, also copy the hidden file STACKER2.BIN from the root directory of the uncompressed drive to the root directory of the disk in the A: drive. You may use the Windows File Manager or another third-party utility to copy the file. If these are not available, do the following:

a. Go to the Stacker directory and type STACKER. Locate the line that shows drive C:. It will look something like:
Drive C was drive C at boot time [D:\STACVOL.DSK = 123.4MB]
b. Note the drive letter in brackets, in this case drive D:.
c. From the DOS directory, type ATTRIB -S -H -R DRIVE:\STACKER2.BIN where drive: is the drive letter you noted above.
d. Copy STACKER2.BIN to the A: drive.
e. From the DOS directory, type ATTRIB +S +H +R DRIVE:\STACKER2.BIN

4. Label the disk "Stacker Startup Disk" and put it away.
Note that you may also copy other files to the disk that may be useful in emergency situations, such as CHKDSK, ATTRIB, SYS, FDISK, FORMAT, EXPAND, and Stacker's ED.EXE.

To create a Stacker Startup disk for all other DOS versions:

1. Insert a blank floppy disk in to drive A:.
2. At the DOS prompt, type FORMAT A: /S <enter>. Follow the instructions on your screen.
3. After the system is transferred to the formatted disk, restart your computer with the disk still in the A: drive.
4. After the computer boots up to the A: drive, type C: and then type CD\STACKER
5. Type CONFIG A:
6. Answer Yes to change the STACKER.INI and CONFIG.SYS files. This transfers the Stacker information onto the disk and makes it Stacker-aware.
7. Label the disk "Stacker Startup Disk" and put it away.

How do I add a new hard disk to a machine on which Stacker is installed?

To add an additional hard disk drive to the system, where the new drive will not be the boot drive, simply follow the hard drive manufacturer's instructions for installation. Most hard drives are recognized by DOS automatically. DOS assigns drive letters at system startup as needed. Hard drives have priority in drive letter assignment, so they get drive letters before Stacker. Therefore, the drive letter of your Stacker drive is moved down the alphabet by one letter for each hard drive/partition that is added.

For example, if you have a single hard drive C: that you compressed with Stacker, an additional drive letter D: is assigned by DOS. The drives are swapped to keep your configuration consistent with the way it was before Stacker, ie, drive C: swaps with drive D:. If you install a new hard disk, DOS assigns it drive letter D:. DOS then assigns the Stacker drive the next available drive letter, E:. Then drive C: swaps with drive E:. This reassignment is automatic.

In order to ensure that your new hard disk is properly installed before formatting it, run DOS's FDISK program. Do not partition your drives at this point. Note whether you have an option to change fixed disk drives. If you do, display the new drive. If it is accessible, yet is 100% unused, the drive is available to partition and then format. If it is not, it may not be properly installed. See the manufacturer's documentation to finish the installation before proceeding with an FDISK partitioning or FORMAT procedure.

If the new drive will be installed as the boot drive, ie, the C: drive, proceed as follows.

Follow the manufacturer's instructions for installation. In order to ensure that your new hard disk is properly installed before formatting it, run DOS's FDISK program. Do not partition your drives at this point. Note whether you have an option to change fixed disk drives. If you do, display the new drive. If it is accessible, yet is 100% unused, the drive is available to partition and then format. If it is not, it may not be properly installed. See the manufacturer's documentation to finish the installation before proceeding with an FDISK partitioning, making the partition active, or FORMAT procedure.

Install DOS to the new drive as you normally would.

After you restart your system, you will not have access to your Stacker drives. Do the following to restore them:

1. Change to the new drive letter of the original drive. For example, if your original drive is now drive D:, go to that drive.
2. Change to the STACKER directory and type CONFIG. Answer yes to the changes that the program wishes to make.
3. Insert Stacker Disk 1 into its drive, and go to that drive. Type \tools\redbl c:.
4. Copy the file STACKER2.BIN from Stacker Disk 1 to the root directory of Drive C:.
5. Restart your system to gain access to the Stacker drives.

You may now compress your new hard drive if you wish.

Stacker 4.0

Some hard disks such as "hard cards" require a device driver statement in the CONFIG.SYS file. Which drive letter the new drive receives depends on whether it is loaded before or after the Stacker lines in the CONFIG.SYS. The syntax of the Stacker lines in your CONFIG.SYS depends on whether Stacker preloads. Examine your CONFIG.SYS file by typing ED /C <enter> at the DOS prompt. If the Stacker lines look like these, Stacker is preloading:

```
DEVICE=C:\STACKER\DPMS.EXE
DEVICEHIGH=C:\STACKER\STA
CHIGH.SYS
```

If the Stacker lines look like these, Stacker does not preload:

```
DEVICE=C:\STACKER\STACKER
.COM
CEVICE=C:\STACKER\SSWAP.C
OM
```

In either case, the drive will receive a letter before Stacker if its statement is before the Stacker lines; and it will receive a letter after Stacker if its statement is after the Stacker lines. You may move the statement to the other side of the Stacker lines if you wish to manage this configuration yourself.

If you have a CD-ROM drive that came in as a later drive letter, say E: or F:, you may have to manually change its assignment. See your CD-ROM documentation for details.

How do I recover from a "Not a Stacker Stacvol file-NOT MOUNTED" or a "Invalid # reserved sectors (must be one) - NOT MOUNTED" error message?

Each Stacker drive is actually a STACVOL.xxx file that is stored on the uncompressed drive. This file consists of a header that contains control information about how and where your data is stored, and a data area that contains all of your compressed data. If you receive one of the messages above, the header is damaged. However, Stacker saves a copy of the header for each Stacker drive. You may use this saved header to restore your STACVOL file.

These instructions assume that the Stacvol file on drive C: is the one that is damaged. If the damaged file is on another drive, substitute that drive letter for C:.

Stacker frequently saves header information. When you start your computer, CHECK /WP saves the header as STACSAVQ.nnn; and when you start Windows or use the CHECK tool, the header is saved as STACSAVE.nnn. You must determine which of these files is newer.

NOTE: Do not use these procedures if you have run SDEFRAG (the Stacker Optimizer), and you have neither run CHECK nor successfully restarted your computer afterwards. In this case, you should contact Stac's technical support department directly for assistance.

1. Start your computer, using a floppy disk if necessary.
2. At the DOS prompt, change to the root directory of your C: drive and type DIR STACSAV*.* /AH. If your version of DOS does not support use of the /AH switch to view hidden files, you must use a third-party utility to see them.
3. Notice whether the file with the latest date and time is STACSAVE or STACSAVQ.

Now to restore the header.

1. From the root directory of drive C: type: DIR /AH. Locate the file STACVOL.xxx. Record the exact spelling of this STACVOL file name.
2. You must clear the system, hidden, and read-only attributes from the STACVOL file. Place your Stacker Disk 2 in drive A: or B:, go to that drive, and type SATTRIB -S -H -R C:\STACVOL.XXX. For STACVOL.XXX enter the spelling of the STACVOL file you recorded above.
3. Insert Stacker Disk 1 into the drive. Make sure you are still logged on to that drive.
4. If the latest saved header as determined in the first procedure is STACSAVE, type \TECH\REPAIR /F C:\STACVOL.xxx where the xxx is the file extension you determined in step 1 above. If the latest saved header is STACSAVQ, type \TECH\REPAIR /F C:\STACVOL.xxx /Q where the xxx is the file extension you determined in step 1 above.
5. Remove all disks from the floppy drives and restart your computer.

How do I increase the size of Stacker drive?

If you decide that you wish to grow your Stacker drive, you may do so from the Stacker Toolbox. Note that the Stacker drive is actually a large STACVOL file on the hard disk, and growing a Stacker drive entails growing this file. In certain configurations you may find that you are unable to grow a drive. This section will explain those circumstances and provide work-arounds.

You may use the Stacker Toolbox in either Windows or DOS to change your Stacker drive size. Full details are contained in Chapter 4 of the Stacker User's Guide.

There are circumstances where you may not be able to grow your Stacker drive:

1. If you have already grown your Stacker drive in the past. A Stacker drive can only be grown to twice its original size. If you try to grow it any more you will receive an error message. In this case the only solution is to uncompress the drive, and run Setup again.
2. If your Stacker drive has been updated from an earlier Stacker version. Run HCONVERT. You will need enough uncompressed space on the drive for the HCONVERT program to temporarily store a copy of the STACVOL header. Note that some updated Stacker drives may be limited by DOS from growing. To use HCONVERT, do the following:

a. Exit Windows, if necessary. At the DOS prompt type STACKER. You will see a display something like this:

```
Drive C was drive C at boot time
[ D:\STACVOL.DSK = 125MB ]
Drive D was drive D at boot time.
```

Note the drive letter in brackets, drive D: in this case. That is your uncompressed drive. The Stacker drive is drive C:. Also note the size of the STACVOL file, in this case 125 MB. You will be growing the STACVOL file later to a larger size.

b. Make sure the drives have no errors.
   Run Stacker's CHECK utility against the Stacker drive, and CHKDSK against the uncompressed drive. If there are errors on either drive, run the appropriate utility again, but add the /F parameter to the command.
c. Type HCONVERT /C D:\STACVOL.DSK /=MG=XXX where D: is the drive letter of the uncompressed drive, and XXX is the desired size of the STACVOL file. To determine the desired size of the file, decide how big you wish your Stacker drive to be. Divide this by 2.5 (which is the default Expected Compression Ratio.) Round to the nearest whole number. This gives you the approximate size of the STACVOL file that you will need. Substitute the result for the XXX above.
d. After HCONVERT finishes, use the Stacker Toolbox to change the size of your Stacker drive.

3. If you have previously shrunk your Stacker drive, the Expected Compression Ratio (ECR) may be out of alignment with the Actual Compression Ratio (ACR). You may not be able to grow the Stacker drive. If so, open the Stacker Toolbox in DOS. Select Expected Compression, and then the drive letter you are working with. Set the ECR to match the ACR. After the operation completes, use the Toolbox again to change the Stacker drive size.

My users seem to be having trouble with the PKZIP file compression utility after installing Stacker 4. What's the problem?

You may experience difficulty using PKZIP 2.04 and Stacker 4.0 if you have a 386 or better computer, have at least 1 MB of RAM, and are using preloaded compression (DOS 6 or higher). The difficulty is caused by the way in which PKZIP uses DPMI services. Use the -) option on the command line when using PKZIP. If you wish to permanently disable PKZIP's use of DPMI, use this option in your PKZIP.CFG file. See the PKZIP documentation regarding making changes to the PKZIP.CFG file.

How do I back up a Stacker drive?

Back up your compressed drive(s) as you normally would. If you have left any of your drives uncompressed and have installed programs or data on them, you should back them up also. You do not need to back up the small uncompressed drives that are created when you compress your drives. In order to determine which drives are the uncompressed "host" drives, go to the DOS prompt and type STACKER. You will see a display something like:

```
Drive C was drive C at boot time
[ D:\STACVOL.DSK = 102.3MB]
Drive D was drive D at boot time
```

In this example, the C drive is compressed and the D drive is its uncompressed physical drive. Back up the C drive and not the D drive.

Do I back up the STACVOL file?

Your backup software may display a large hidden file on your uncompressed drive called STACVOL.xxx. This file is the file that contains all of your compressed data. When you back up the Stacker drive, you have already backed up the contents of the STACVOL file, and you do not need to back it up again.

Are the backed up files compressed?

Stacker uncompresses files as they are read from the hard drive. They are then handed to the backup software in their original, uncompressed state.

If the backup software uses compression, it will recompress the files before storing them to diskette or tape. This means that the files can be restored to any drive, compressed or not.

How do I restore my backup?

Restore your backup to any drive in the same way that you normally would. You should remember that your backed up data is uncompressed. If you restore it to a drive that does not have Stacker running, it will also be uncompressed on the target hard drive. This means that if you have had a major disk crash and are restoring your backup to that drive, you must first run the Stacker SETUP on that drive in order for the restored backup to be compressed. This may be important if the drive is not physically large enough to hold all of the backed up data.

How do I ensure that Stacker is installed in the most efficient way, using as little conventional DOS memory as possible?

The Stacker device driver may be loaded into the UMBs if the system is configured properly and if there is a UMB large enough to accommodate it. Freeing up UMBs that are large enough may require much trial and error. You may have to experiment with the order in which device drivers are loaded. If you attempt to load the Stacker driver into the UMBs and they are not large enough, the driver will simply load into the lower 640 KB of conventional memory.

PCSA

Acknowledgement

This article has been prepared by PCSA's staff, from application notes and technical bulletins issued by Stac Electronics.